Maia by matillion

Document Classification: PUBLIC

WHITEPAPER

AI Security Whitepaper for Matillion Maia Foundation

Version 2.1 | 2026 - 10 - 22

How Matillion designs, secures and governs the AI capabilities in the Maia platform and Maia Team.

1. Purpose and scope

Matillion Maia embeds AI directly into the work of building and operating data pipelines. That places AI in the path of customer data, which raises questions that conventional platform security documentation does not answer. This paper answers them.

It covers the AI capabilities Matillion operates on a customer's behalf, the components that let a customer bring their own AI services into a pipeline, and the controls that apply to both. It is written for the people who assess Matillion as a supplier.

Controls that are not specific to AI, including network architecture, secure development, vulnerability management, business continuity and physical security, are covered in Matillion's platform security documentation and in the audit reports available through the Matillion Trust Center. This paper does not repeat them.

Where a control belongs to the customer rather than to Matillion, the paper says so. Section 13 consolidates those into a shared responsibility model.

2. Design principles

Five principles shape how AI is built into Maia. The controls described in the rest of this paper follow from them.

- **Authority stays with the customer.** Maia proposes; people decide. Actions that touch data or change state require approval from a user, and Maia operates inside the permissions that user already holds. Automation is available, but only where a customer has deliberately configured it.
- **Capability is bounded by design.** Agents act through a defined set of tools. An agent cannot perform an action for which no tool exists, so the consequence of any failure, including a successful attack, is limited to operations Matillion has deliberately exposed and the customer has permitted.
- **Inference, not training.** Customer data is sent to a model to produce an answer and is never used to train, fine-tune or adjust the weights of any model.
- **Isolation is absolute.** No context, cache, index or store is shared between customers. Tenancy isolation in the AI path is enforced by the same mechanisms that govern the rest of the platform.
- **Output is reviewable before it takes effect.** Maia produces graphical pipelines, not opaque code. Every change it proposes is visible, amendable, and held as uncommitted work until a person commits it.

3. What Maia does

Maia is a set of capabilities rather than a single feature. Each has a different relationship to customer data.

Capability	What it does
Maia Team	A conversational interface in the Designer, used to author pipelines and automate DataOps tasks. It also encompasses specific actions on components (e.g. generate SQL expressions), as well as notes/documentation generation. Agents are modelled on data team roles and collaborate on a request.
Maia for Custom Connectors	Assisted authoring of custom connectors.
Maia Root Cause Analysis	Diagnostic assistance when a pipeline fails.
Context Engine	A knowledge graph and vector index that grounds Maia's reasoning in the customer's own data estate. Covered in section 7.
AI pipeline components	Components a customer places in a pipeline to call an AI service they select and control. Covered in section 12.

The first four are operated by Matillion under the controls in this paper. The fifth is configured by the customer, who chooses the provider, holds the credentials and determines what data is sent.

3.1 What Maia does not do

- It does not use customer data to train, fine-tune or benchmark any model.
- It does not copy the customer's data warehouse into Matillion's infrastructure.
- It does not expose secret values to a model or write them to telemetry.
- It does not apply a pipeline change without a person committing it or setting permissions in this manner.
- It does not hold permissions beyond those granted to the identity that invoked it.
- It does not pass model inference costs to the customer.
- It does not operate at all in an account where the capability has been switched off.

4. Architecture

4.1 Components

Control plane. Operated by Matillion. It hosts the Designer, the orchestration layer that coordinates agents and calls the model, the database holding conversation state, the Context Engine backend, and the observability platform. Conversation history in the database is encrypted with a dedicated data encryption key.

Model provider. Maia's reasoning is powered by models accessed through Amazon Bedrock, invoked from Matillion's own AWS account under Matillion's commercial agreements. Data is transmitted for inference and is not retained by the provider or used for training.

Data plane. Pipelines execute and data is accessed through runners. Runners connect to the warehouse to retrieve schema information and samples, to validate pipelines and to execute them. Runners are the only component that can resolve a secret reference to its value. Matillion offers both Matillion-hosted runners and runners the customer deploys into their own Kubernetes cluster on Amazon EKS, Google GKE or Azure AKS.

Customer systems. The data warehouse holds the customer's data and is not replicated into Matillion's infrastructure. Pipeline definitions are version-controlled in Git, either a repository the customer provides or one Matillion hosts.

Connections between the client and the control plane, and between the control plane and the model provider, use HTTPS with TLS 1.2 as the enforced minimum.

4.2 What reaches the model

Only the following classes of data are placed in a model's context.

Class	Detail
User prompts and Maia responses	The conversation, including prior turns, which are re-sent on each turn so the model retains continuity.
Context files and skills	Markdown content adding extra context about a behaviour Maia has to enforce when building pipelines (e.g. naming conventions, standards) and extra business context to inform the pipeline generation.
Attachments	Multi-modal attachments added to the Maia prompt.
Schema and metadata	Table and column names and data types.
Data samples	A bounded number of rows from a source table, subject to row-count and payload limits. *Sampling can be switched off, in which case no customer records reach the model.
Pipeline definitions	The pipeline being authored or diagnosed.
Environment references	The names of environments and environment defaults. Configured values are not included.
Product documentation	Product documentation (using a lookup to our documentation website and to an allow-list of partners/platforms we integrate to) to add extra information about graphical components and use cases.

Class	Detail
Knowledge graph content	Knowledge graph information to infer the relationships between business concepts and physical data assets.
Operational errors	Errors an agent encounters while performing an action, such as a query timing out, so that it can correct itself.

Secret values are never included. Data samples are the most sensitive class and the only one a customer can remove from the AI path independently of the capability itself. Sampled rows persist in conversation history in the Maia database, and may also appear in observability traces.

Data residency. Customers select the region in which their Maia instance operates: United States, European Union or Australia. The control plane, the Maia database including conversation history, and the Context Engine with its vector index all reside in the selected region. Model inference for United States, European Union and Australia instances remains within that territory, in whichever regional data centre the model provider selects. For the United Kingdom, no UK-only models are currently available to Maia, so requests from UK instances are served from within the European Union, selected by the provider. Matillion cannot pin inference to a single data centre today. Two components sit outside the selected region: Matillion's Hub and billing databases are global and hold user profile information including email addresses, and the observability platform described in section 8.5 is hosted in Ireland for all customers whatever region they select. Current published detail is maintained in the Maia documentation at docs.maia.ai.

5. Human oversight and automation

Most security reviews of an agentic product reduce to one question: what can it do without a person present? Maia's answer has three levels.

Level	Behaviour	Where it applies
Assisted	Maia proposes a change. A person reviews it and commits it.	Pipeline authoring, custom connector authoring, root cause analysis. This is the default.

Level	Behaviour	Where it applies
Supervised	Maia requests approval for a specific action, and acts only once the user grants it.	Sampling data, running a query, executing a pipeline, deleting an object.
Customer-configured automation	A task proceeds without a person present at the moment of execution.	Available only where a customer has explicitly configured a task to run this way, and only within the permissions of the identity the task runs under.

Automation is opt-in. Maia does not acquire it by default and cannot enable it for itself. A customer who never configures it retains a person in the loop for every action Maia takes. A customer who does configure it is making a deliberate delegation, bounded by the permissions granted to the identity concerned.

Alongside these levels, two controls apply at all times. A user can interrupt Maia at any point and take over manually. And because Maia writes pipelines as graphical artefacts, the user sees exactly what has been proposed before anything is committed.

6. Identity, access and authority

6.1 Three layers of access control

Account. Maia can be switched off for an entire account. When it is off, no user in that account can reach any AI capability, whatever their individual permissions. Customers with a policy position against AI in their data platform can hold that position while using Matillion.

User. Where Maia is available to an account, access is a permission granted or withheld per user through Matillion Hub as part of the user's role.

Action. Where both account and user permit it, Maia still requests explicit approval before an action that reads data or changes state. Sampling data, running a query, executing a pipeline and deleting an object each require the user to agree.

These controls behave identically in trial, evaluation and paid accounts. There is no relaxed model for non-production use. Actions are audited through the main audit service (e.g. when a user approves a Maia Team action, submits prompts, or even switches on "bypass permissions")

6.2 Identity and permissions

Every identity in Matillion is governed by the same role-based access control model, whether it is a person, a machine user or a service identity. Custom roles will allow a customer to decide precisely how much access any identity holds.

Maia acts under the identity that invoked it and inherits that identity's permissions. It holds no standing privilege of its own. Enabling Maia therefore grants no access beyond what the invoking identity already had, and the customer's existing least-privilege model continues to apply.

This extends to programmatic use. Where a customer creates tasks through the Agent Tasks API, the credentials are associated with a service identity, and that identity is scoped through the same role model. A customer restricting Maia's reach, for example keeping it away from a production environment, does so by scoping the permissions of the identities that invoke it.

6.3 Secrets

The architecture prevents a model from ever seeing a secret value.

- The control plane holds and passes secret *references*, not values.
- Only data plane runners can resolve a reference to its value.
- The flow is one-way. Maia can ask that a secret be used in an action the runner performs; it cannot retrieve the value.

Secret material does not enter model context and is not written to telemetry. Credentials for customer-configured AI services remain the customer's own, held in the platform's secret management, and their rotation remains under customer control.

For runners the customer deploys into their own Kubernetes cluster, secret storage is part of the cluster configuration the customer owns. Matillion's deployment guidance for EKS, GKE and AKS sets out the recommended pattern, which is a managed secret store such as AWS Secrets Manager, Google Cloud Secret Manager or Azure Key Vault integrated through the External Secrets Operator, providing centralised, auditable secret management with rotation support. Customers deploying their own runners should follow that guidance rather than relying on platform defaults.

7. The Context Engine

Maia produces better results when it understands the customer's data estate rather than reasoning from a single prompt. The Context Engine provides that understanding. It builds a knowledge graph and a vector index over sources the customer nominates.

The customer decides what is indexed. The Context Engine does not discover and ingest on its own initiative. It indexes what the customer tells it to index. Today that is warehouse schemas and pipeline execution data; support for external connections and unstructured documents is in development. A customer who nominates nothing has an empty Context Engine.

What it holds. Table and column names with their data types; logical entities and table summaries derived from the sources indexed, which may include data values drawn from sampled rows; attributes such as KPI definitions drawn from the customer's pipelines; and any documents the customer uploads deliberately.

Storage and encryption. The knowledge graph is held in Amazon Neptune and is encrypted at rest. Content associated with the graph, together with the vector index, is held in a database Matillion operates and is encrypted at rest. Matillion is implementing envelope encryption for that content, under which content is encrypted before it is written to a store that is itself encrypted, so that access to the store alone does not yield readable content. Data in motion during indexing, including staging while sources are processed, is encrypted.

Isolation. The graph and index are subject to the same tenancy isolation as the rest of the platform. Nothing is shared between customers.

Deletion. Context Engine data is deleted on request, and on exit.

8. Data protection

8.1 Encryption

Data is encrypted in transit throughout, over HTTPS with TLS 1.2 as the enforced minimum. This covers traffic between the client and the control plane, between the control plane and the model provider, and during Context Engine indexing.

Data at rest is encrypted using AWS Key Management Service. Keys are rotated annually. Conversation history carries an additional data encryption key, and Context Engine content is moving to envelope encryption as described in section 7.

8.2 Tenancy isolation

Isolation is enforced in three layers, consistently across the platform and the AI path. Every request carries an authenticated token from which the user and their account are verified. Tenancy is built into the data structures themselves, so that every record carries the account and project it belongs to and every query is bounded to the correct account. Role-based access control then prevents a user reaching content that is not theirs.

No cache, index, embedding store or knowledge graph is shared across customers.

8.3 Retention and deletion

Data	Retention
Conversation history and associated stored context	Two years from last activity, after which it is deleted from the database.
Database backups containing conversation data	Production database snapshots are retained for two to three years, encrypted, and copied out of region where possible. Backups are not anonymised, so a restore is attributable to the customer.
Session data, including samples and metadata held for an active session	Two years from last activity, after which it is deleted from the database.
Prompts and completions at the model provider	Not retained.
Context Engine graph and index	Maintained while the nominated source remains in scope. Deleted on request and on exit.

Telemetry traces supporting service monitoring are retained for 30 days in the EU region, as described in section 8.5. Retention is set to support the customer's obligations under GDPR, CCPA and equivalent regimes. Full retention schedules are available on request. Customers can request deletion of AI-related data, including Context Engine content, through their normal support channel. Deletion applies to live systems. Matillion does not currently operate a mechanism to remove an individual customer's records from encrypted database backups, which expire on their own retention schedule.

8.4 Model training and service improvement

No customer data trains any model. All interaction with a model is inference. Customer prompts, conversations, samples and pipelines are not used to train, fine-tune or adjust the weights of any model, and Matillion builds no derived models from them. Matillion's agreements with its model provider prohibit the use of transmitted data for training.

Matillion does use its own operational understanding of how Maia performs in order to improve it, and is explicit about what that means. Prompts and conversations inform Matillion's understanding of what users are trying to achieve, measurement of how well Maia performs against that intent, and the design of test scenarios. This work does not train models. Access to raw Maia responses, which may contain sampled customer data, is restricted and granted only through a controlled break-glass process.

Matillion's evaluation programme, which measures Maia's quality and gates its releases, runs against synthetic datasets built by Matillion engineers. It uses no customer data.

Access to this material is limited to the purposes described above and is governed by the controls in section 8.5.

8.5 Observability and Matillion personnel access

Matillion operates an observability platform, self-hosted within Matillion's own environment, which captures traces of requests and responses so that the service can be monitored, evaluated and supported. Trace data is also processed in Matillion's analytics platforms, to support the service improvement work described in section 8.4.

Access to the production observability environment is controlled. It is requested through Matillion's access management system, granted under a documented data usage agreement that sets out the responsibilities attaching to the data and its sensitivity, and logged. Access follows Matillion's standard access management process, which is defined by the Site Reliability Engineering function and examined at audit. Traces are retained for 30 days and then expire, and the platform is hosted in the EU region.

Personal data is not deliberately captured in telemetry. Where it is found to have reached the platform, Matillion's security incident management process is triggered and the affected data is deleted.

9. Output integrity

A wrong answer in a chat product is a wrong answer. A wrong transformation in a data pipeline propagates into everything downstream of it. Maia is built accordingly.

9.1 Validation

Generated output is validated automatically against three checks: schema validation, SQL validation and component validation. Where a check fails, the error is returned to the agent, which corrects its output, and the same error is surfaced to the user in the conversation. A failure is therefore visible rather than absorbed silently. Self-correction is bounded; an agent does not retry indefinitely.

9.2 Nothing takes effect until a person commits it

Changes Maia makes to a pipeline are held as uncommitted work. A user must commit them to a branch, in the customer's own repository or in Matillion-hosted Git, before they form part of the pipeline. Two consequences follow. An unwanted change can be discarded before it ever reaches version control. And once committed, it is reversible by the same means as any other change, through a standard revert or branch reset.

9.3 Bounded consequence

The reach of a change a user approves is bounded by the permissions of the identity that invoked Maia, the grants on the connection in use, and the environment that identity can reach. Maia does not modify connections, credentials or environment configuration, and does not act on systems outside the environment it is operating in. Where a generated pipeline contains a destructive operation, Maia identifies it and the user must approve it explicitly before it proceeds, unless the customer has configured the task to run without that approval. Whether a generated pipeline is safe to re-run is a property of how the customer has asked for it to be built, and Maia will build idempotent pipelines when instructed to. Changes to pipelines are recorded in the platform audit trail.

10. AI-specific threats

Language models introduce failure modes that conventional software does not have. Matillion maintains a documented threat model for Maia covering the actors, surfaces and assumptions relevant to it. Consistent with normal practice, the detail of that model is not published.

10.1 Prompt injection

Prompt injection, where instructions hidden in content subvert a model's intended behaviour, is the most significant novel risk in an agentic system. Content placed in a model's context that originates outside Matillion's control is untrusted by nature. That includes rows sampled from a warehouse, schema metadata, documents a customer uploads and results retrieved from external sources.

Matillion's defence is layered and rests primarily on architecture rather than on filtering, because architecture holds even when filtering fails.

- **Bounded tool surface.** A successful injection cannot cause Maia to perform an action for which no tool exists.
- **Approval gate.** Actions with real-world consequence require the user to agree. An injection cannot silently trigger them.
- **Inherited permissions.** Maia operates within the invoking identity's permissions. Injection cannot escalate privilege.
- **Uncommitted output.** A change must be reviewed and committed by a person before it takes effect.
- **Prompt construction.** System instructions, user input and retrieved content are structurally separated when a prompt is assembled.
- **Screening of retrieved content.** Matillion operates its own prompt-injection screening over content retrieved from external sources before that content reaches a model.
- **Model-level restrictions.** The underlying models decline or moderate responses to requests that appear harmful.

10.2 Agent behaviour

Tool output is parsed and validated before it re-enters model context. Agents are isolated from one another, so one agent cannot interfere with another's execution. Agents can chain several tools before returning to the user, and that chaining is bounded by internal limits; a user can stop an agent at any point.

10.3 Supply chain

Models and the libraries in the agent execution path are vetted and version-pinned, managed centrally by Matillion's AI engineering function. Documentation search within the product is provided with assistance from a third-party service. Customer data is not passed to it; it receives queries against Matillion's published documentation.

10.4 Cost

Matillion bears the cost of model inference for Maia. Customers do not incur token charges, so a repeated or looping agent operation carries no financial exposure for the customer.

10.5 Resilience

Maia is built to fail safe. A model provider outage does not cascade into pipeline generation failures, and if Maia is unavailable the platform continues to operate. Every task Maia performs can be performed manually, so the loss of AI assistance costs productivity rather than capability.

10.6 Testing

Matillion commissions an annual penetration test and operates a continuous bug bounty programme. The penetration test report is available through the Matillion Trust Center.

11. Model and capability lifecycle

11.1 Model selection and change

Models are selected, vetted and version-pinned by Matillion's AI engineering function. Updates are validated before deployment, and versioning provides a path back if a regression is found. The model or provider behind Maia may change as the capability develops. Matillion notifies customers where such a change alters the architecture or data residency position described in its published documentation.

11.2 Evaluation

Matillion operates an evaluation programme for Maia, owned by a dedicated team. Its purpose is to quantify the accuracy and running costs of Maia. Evaluations run Maia against standard test datasets under varying conditions. Those datasets are synthetic and contain no customer data.

Results are used to gate releases: a capability must meet quality thresholds before it reaches customers, in the same way functional testing gates any other release. A material regression identified by an evaluation is handled as a software regression, entering the defect workflow with customer communication proportionate to severity. Production behaviour is monitored through continuous online evaluation, owned by engineering.

11.3 Development lifecycle

AI capabilities are built under Matillion's secure development lifecycle, which is documented and examined as part of Matillion's ISO 27001 and SOC 2 programmes. Matillion has additionally developed an AI-specific development lifecycle, incorporating AI-assisted checks in code development and in test and release.

12. Customer-configured AI components

Separately from the capabilities Matillion operates, Matillion provides pipeline components through which a customer integrates an AI service of their own choosing. These support loading unstructured data, calling language models, building retrieval-augmented generation pipelines and writing to vector stores, and integrate with services including Amazon Bedrock, Azure OpenAI, Google Gemini, Pinecone, pgvector, Snowflake Cortex and Databricks AI.

In this model the customer selects the service, holds its credentials, determines what data is sent and bears the cost. Data transmitted to a service the customer has selected passes outside Matillion's control, and assurance of that service rests with the customer. Matillion provides the integration and the control points around it: platform secret management for credentials, role-based access control over who can edit pipelines, and the ability to restrict the availability of AI components at account level. Where a pipeline stages data in Amazon S3, the staging components offer a choice of server-side encryption with a key held in AWS KMS, server-side encryption with an S3-managed key, or client-side encryption with a customer key. Encryption of staged data is a customer configuration decision. Customers handling sensitive data should select one of these options rather than leaving staging unencrypted.

13. Shared responsibility

Control	Matillion-operated Maia	Customer-configured AI components
Model selection and provider agreement	Matillion	Customer
Assurance of the AI provider	Matillion	Customer
Cost of inference	Matillion	Customer
Credential management	Matillion	Customer, using platform secret management
Credential rotation	Matillion	Customer

Control	Matillion-operated Maia	Customer-configured AI components
What data reaches the model	Matillion, bounded by permissions, connection grants and the sampling setting	Customer, determined by pipeline design
Masking or minimisation before transmission	Matillion, bounded by the sampling control	Customer, using platform transformation components
Injection screening and content safety	Matillion	Customer, configurable at their chosen provider
Transport security	Matillion	Shared. Matillion secures connections it initiates; the customer controls the endpoint
Who may use the capability	Matillion provides account, user and action-level controls; the customer operates them	Shared, through account-level restriction and pipeline permissions
Which environments are reachable	Customer, through the permissions of invoking identities	Customer
Warehouse-side least privilege	Customer	Customer
Encryption of data staged in S3	Customer, per component	Customer, per component
Kubernetes configuration for customer-deployed runners	Customer	Customer
Recovery of data written by an incorrect pipeline run	Customer, through warehouse backup and time travel	Customer

14. Governance and compliance

14.1 Accountability

AI risk in the product is owned by Matillion's Chief Technology Officer and governed through Matillion's standard Risk Management Forum, which is the same governance route used for other material technology risk. AI risk is not managed in a separate structure with separate standards.

14.2 Certifications

Matillion's ISO 27001 certification and SOC 2 attestation cover the Maia service and the infrastructure it runs on. Maia is not outside the certified boundary.

Matillion is preparing for certification to ISO/IEC 42001, the management system standard for artificial intelligence, and is targeting certification in the first quarter of 2027. Matillion is not certified to ISO/IEC 42001 today.

Certifications, audit reports, the Record of Processing Activities and the penetration test report are available through the Matillion Trust Center at matillion.com/trust-center.

14.3 Framework alignment

Maia's controls are developed with reference to the OWASP Top 10 for Large Language Model Applications and the NIST AI Risk Management Framework. The table below shows where this paper addresses each area of the OWASP list.

Risk	Addressed in
Prompt injection	10.1
Insecure output handling	9.1, 9.2, 10.2
Training data poisoning	8.4, 10.3, 11.1
Model denial of service	10.4, 10.5
Supply chain	10.3, 11.1
Sensitive information disclosure	4.2, 6.3, 8.1, 8.2, 8.5

Risk	Addressed in
Insecure tool design	2, 10.2
Excessive agency	5, 6, 9.2, 9.3
Overreliance	9.1, 9.2
Model theft	Not applicable. Matillion hosts no proprietary model weights.

14.4 Incident management

AI incidents are handled within Matillion's security incident management process, which is documented, audited and reviewed annually. Matillion recognises AI-specific incident classes alongside conventional ones, including output that causes data integrity harm, adversarial input, unintended disclosure, model or provider compromise, capability regression and an agent acting outside its intended authority.

Maia can be switched off immediately at account level, which serves as the containment control of last resort.

14.5 Sub-processors

Model inference and the Context Engine's graph store are provided by Amazon Web Services under Matillion's agreements. The observability platform is self-hosted by Matillion. Trace data from it is processed in Matillion's analytics platforms, as described in section 8.5. For more details, refer to our DPA at https://www.matillion.com/legal/data-processing-agreement, and Annex III – List Of Sub-Processors mentioned at the bottom of DPA.

15. Customer controls

The controls a customer can operate, and what each costs in capability.

Control	Scope	Effect	Capability cost
Switch Maia off	Account	No user in the account can reach any AI capability. Also the emergency stop.	All AI assistance. Manual authoring is unaffected.
Restrict AI pipeline components	Account	Components are unavailable in the account.	No AI service integrations in pipelines.
Switch off sampling	Account	Maia works from metadata only and does not see customer records.	Reduced accuracy where the shape and content of data would inform a result. Without it, Maia creates pipelines based on metadata, with no guarantee on the outcome quality.
Reduce what reaches telemetry	Account	Switching off sampling also keeps customer records out of the traces captured in observability, because those records never enter the conversation in the first place.	As for sampling above.
Choose what the Context Engine indexes	Account	Only nominated sources are indexed.	Less grounding, and therefore less accurate context, for un-indexed areas.

Control	Scope	Effect	Capability cost
Withhold the Maia permission	User	A named user has no AI access; others are unaffected.	AI assistance for that user.
Scope identity permissions	User or service identity	Bounds the environments, projects and objects Maia can reach when invoked by that identity.	Maia cannot assist with what the identity cannot reach.
Decline an action	Single action	The action does not proceed.	That step is completed manually.
Interrupt Maia	Session	Maia stops and the user takes over.	None. Work in progress is preserved for review.
Discard uncommitted changes	Pipeline	Proposed changes are abandoned before reaching version control.	None.
Revert a committed change	Repository	Standard Git revert or branch reset.	None.
Select staging encryption	Component	Encrypts data staged in S3.	None. Recommended for sensitive data.
Restrict warehouse grants	Data platform	Bounds everything Maia can read or write.	Maia cannot assist with objects it cannot see.
Request deletion	Account	AI-related data, including Context Engine content, is deleted.	Loss of accumulated context and history.

Several of these are applied by Matillion on request rather than through self-service configuration. Customers should raise them through their Account Executive or support channel.

16. Further information

- Matillion Trust Center, for certifications, audit reports, the Record of Processing Activities and the penetration test report: matillion.com/trust-center
- Maia product documentation, including deployment guidance: docs.maia.ai